Accurate estimates of false alarm number in shape recognition

Pablo Musé, Frédéric Sur, Frédéric Cao, Yann Gousseau, Jean-Michel Morel

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Accurate estimates of false alarm number in shape recognition

Pablo Musé, Frédéric Sur, Frédéric Cao, Yann Gousseau, Jean-Michel Morel

N°5086

Janvier 2004

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--5086--FR+ENG

INRIA
RENNES

Accurate estimates of false alarm number in shape recognition

Pablo Musé*, Frédéric Sur*, Frédéric Cao†, Yann Gousseau‡, Jean-Michel Morel*

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Vista

Rapport de recherche n ° 5086 — Janvier 2004 — 28 pages

Abstract: There are many shape recognition algorithms. Their Achilles heel usually is the control of the number of false positive, or false alarms. A match between two shapes $\mathcal{F}$ and $\mathcal{F}'$ being proposed with a distance d, we compute the "number of false alarms" of this match. This number is computed as an upper bound of the expectation of the number of shapes which could have casually a distance lower than d to $\mathcal{F}$ in the database. It turns out that a simple encoding of shape elements as pieces of level lines leads to compute numbers of false alarms for the good matches as small as 10^{-13}. As an application, one can decide with a parameterless method whether any two digital images share some shapes or not.

Key-words: Shape recognition, *a contrario* model, meaningful matches, number of false alarms.

(Résumé : tsvp)

* École Normale Supérieure de Cachan, 61 avenue du Président Wilson, 94235 Cachan Cedex, France, {muse,sur,morel}@cmla.ens-cachan.fr
† IRISA/INRIA, Campus universitaire de Beaulieu, 35042, Rennes Cedex, France, Frederic.Cao@irisa.fr
‡ TSI, ENST, 46 rue Barrault, 75643 Paris Cedex 13, France, gousseau@tsi.enst.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Estimation du nombre de fausses alarmes en reconnaissance des formes

Résumé : Il existe de nombreux algorithmes de reconnaissance de formes. De manière générale, leur talon d'Achille est le contrôle du nombre de faux positifs (ou nombre de fausses alarmes). Pour deux formes $\mathcal{F}$ et $\mathcal{F}'$ à distance d, le présent article propose une définition du «nombre de fausses alarmes» de la mise en correspondance de $\mathcal{F}'$ avec $\mathcal{F}$. On montre que ce nombre est une borne supérieure du nombre de formes qui sont casuellement à une distance de $\mathcal{F}$ inférieure à d dans la base de données. On montre expérimentalement que des éléments de formes simplement codés à partir de lignes de niveau permettent d'atteindre des nombres de fausses alarmes de l'ordre de 10^{-13}. Cela permet de déterminer, sans paramètre, si deux images partagent ou non des formes.

Mots-clé : Reconnaissance de formes, modèle a contrario, mise en correspondance significative, nombre de fausses alarmes.

Contents

1 Introduction **3**
1.1 Probability of wrong match or number of false alarms? 5

2 Meaningful matches and *a contrario* model **6**

3 From images to codes **8**
3.1 Shape extraction and normalization 8
3.2 Modelling 9
3.3 Six close to independent and complete shape features 9

4 The background model **10**
4.1 The background model and a detection terminology 10
4.2 Testing the background model 10
4.2.1 Independence testing 10
4.2.2 Number of detections testing 11

5 Experiments **13**
5.1 Casablanca 13
5.2 Weather 16
5.3 Uccello 20

6 Recognition is relative to the database **23**

7 Discussion about hypothesis testing **24**

8 Perspectives and conclusion **26**

References **26**

1 Introduction

Shape recognition is the field of computer vision which deals with the problem of finding out whether a query shape lies or not in a shape database, up to a certain invariance. This shape database is usually extracted from an image or a set of images. When we refer to "shape database", we therefore refer to sets of images as well, along with a shape extraction algorithm. The purpose of this paper is not to propose a new shape recognition procedure. Our main scope is to define parameterless criteria to decide whether two shapes are alike or not. All the same, we cannot avoid discussing first shape recognition in its general nature. Shape matching decisions indeed widely depend upon the former steps of the involved shape recognition system. Such a system always gives a precise numerical shape representation. Of course, the details of the shape description influence the shape matching criteria.

Shape recognition systems are usually split into three stages.

1. **Features extraction.** Shapes, or shape elements have to be described in a convenient way. In consequence, some features are extracted from the shapes. In order to achieve shape recognition up to a certain invariance, there are mainly two possibilities. Either shape features in themselves are invariant or they are not. In the latter case, the further comparison process has to take invariance into account. Thus, we have to give some details on the various usual shape features. We will indicate those which are best from the invariance and decision making viewpoints.

   a) **Invariant features.** Those invariant features may be computed directly on the image, or after the shape has been extracted, usually as a closed curve or chained points. Features can be differential or integro-differential invariants at some special points (like corners [29]) or regions (*e.g.* coherent regions [5, 34]) of the image. In the case of closed curves, Fourier descriptors [35] or invariant moments [12] can be used. In the case of shapes encoded as curves, Lamdan *et al.* [16], followed by Rothwell [27] have proposed local descriptors of shapes, invariant up to similarity or affine transforms. These features are based on the description of pieces of non-convex curves lying between two bitangent

points. Such features are affine invariant and the use of bitangent lines ensures a lot of robustness to noise. Lisani [19, 18] improved this bitangent method by associating with each bitangent to the shape a coordinate system and defining an affine or similarity normalized piece of curve (we will give more details on the local similarity invariant normalization).

b) **Non-invariant features.** Features can also be made of a set of edges or edgels (a point on an edge with a direction). Groups of features are more informative than individual features, and consequently enhance the matching stages: chained edgels [25] can be considered. Another possibility is to choose "landmarks" by hand [31].

2. **Matching.** This second stage strongly depends on the feature extraction step. If the features are invariant, then comparison between shapes is easy to achieve by using any of the L^p distance, Hausdorff distance, matching distance, or a Procrustean distance [31]. If instead the extracted features are not invariant, the matching procedure must check lots of configurations and therefore becomes computationally heavier. In the latter case the matching process must compare a query shape and all its transformed versions with all shapes in the database. This explains why the matching of non-invariant features often is split into the two following steps: pre-matching (selection of some candidates), then accurate matching (check the candidates features versus the query features). This is precisely the basic structure of Geometric Hashing [17], or of the Generalized Hough Transform [4].

3. **Decision.** This third and last step is, or should be the key point of all shape recognition algorithms. Once two shapes are likely to match, how is it possible to come to a decision? Several authors have proposed probabilistic or deterministic distances, or voting procedures as well, to sort the matches [32]. Now, to the best of our knowledge, the best methods only succeed in ordering the candidates to a match. In other terms, shape recognition systems usually deliver, to any query shape, an ordered list of the shapes which are likely to match with the query. When rejection thresholds exist, they are very specific to a very particular shape recognition algorithm.

Feature detection is obviously unavoidable. Nevertheless, most feature detectors (such as edge or edgel detectors, corner detectors, *etc.*) do not give global structures, but only clouds of points. The provided description is consequently quite rough and the only possible matching procedure is geometric hashing, or similar techniques. However, these methods show serious drawbacks, such as space and time complexity or numerous thresholds which endanger the robustness of the process. Reducing and organizing the mass of information thus appears essential: the higher level the features, the more robust the comparison. For example, chains of edges are much more discriminative than a simple, unstructured collection of edges. Unfortunately, chaining edges is an unstable procedure, depending on several thresholds. This point is very problematic in practice: each step of most shape recognition systems indeed introduces many interdependent thresholds. This makes the acceptance / rejection decision all the more delicate.

To summarize, the alternative is as follows: either, primitive parameterless extraction procedure (edgels) followed by intensive search like geometric hashing: in that case, because of the computational complexity, we can attain all the most rotation-translation invariant recognition. Or, one needs a more sophisticated extraction, followed by a normalization procedure. Now, this is not reliable, being parameter-dependent when edgel-chaining is used.

So, in order to be computationally efficient, shape recognition must rely on a nontrivial shape extraction procedure, yielding significant pieces of shapes. This extraction must, however, be parameterless.

Let us discuss further the shape extraction procedure. Many authors merely assume that shapes are boundaries of well contrasted objects over a uniform background. In this ideal framework, boundaries obtained by edge detection are certainly the most stable and global description of the shapes and can be obtained immediately. Now, this case is pretty particular. In natural images, shapes, *i.e.* silhouettes of objects are not easily extracted. Only very few authors tackle the realistic problem of extracting shapes out of real images. If we are interested in building an automatic decision rule, we should use a parameterless shape extraction providing invariant descriptors.

Lisani's algorithm [18, 19] meets these requirements. It is based on normalized pieces of level lines (with similarity or affine invariance). This algorithm just needs an automatic decision rule, which it is the aim of this article to propose.

Let us specify what we mean by "automatic decision rule" for shape matching. We are looking for a query shape in a shape database. A distance between shapes is available, so that the smaller the distance, the more similar the shapes. The question is: what is the threshold value for that distance to ensure recognition?

Now, given two shapes and an observed small distance δ between them, there are two possibilities:

1. both shapes lie at that distance because they 'match' (correspond to two instances of the same object).
2. there are so many shapes in the database, that one of them matched the query just by chance.

Our aim is to evaluate the probability (or rather the expectation) of the possibility 2 for any δ. If this number happens to be very small for two shapes, then possibility 1, namely a matching decision, must be taken, because this match is not likely to be due to chance.

In order to fix this decision threshold, the distribution of distances between shapes in the database must be learned (in other words the *background* must be modelled). This distribution yields the probability that two given shapes lie at any fixed distance. If a match between two shapes is very unlikely to be due to chance (if the number of false alarms is low) then pairing them is highly relevant.

An obvious objection to what we propose here is following: recognition occurs between two shapes $\mathcal{F}$ and $\mathcal{F}'$ at distance δ when δ is so small that the database could not contain other shapes $\mathcal{F}''$ matching $\mathcal{F}$ at that distance just by chance. We cannot evaluate the probability of this event, since the database yields no examples of such false alarms! So, all the trick here is to be able to evaluate a very small probability, and this cannot be done just by frequency analysis. Let us review and discuss some anterior methods.

1.1 Probability of wrong match or number of false alarms?

Some authors have addressed this question of 'wrong matches' occurring purely by chance, but the proposed models do not lead to an automatic recognition criterion. Let us discuss two interesting examples.

In [25], the authors present a method for automatic target recognition under similarity invariance. Objects and images in which the objects are sought are encoded by oriented edges, and compared by using a relaxed Hausdorff distance. Modelling the matching process by a Markov process leads to an estimation of the probability $P_K(t)$ of a false alarm between K consecutive edges for a given transformation t. The authors give an estimate of the probability of a false alarm occurring over the entire image by computing $1 - \prod_t (1 - P_K(t))$, which is used to take a decision. Let us quote the authors: *"One method by which we could use the estimate is to set the matching threshold such that the probability of a false alarm is below some predetermined probability. However, this can be problematic in very cluttered images since it can cause correct instances of targets that are sought to be missed."*

This methods raises several problems. Finding a false alarm at a given location is clearly not independent of finding it at a close location. Next, the real quantity to be controlled is not the false alarm probability, but the expected number of false alarms.

The authors of [14] are interested in fixing a threshold on the proportion of model features (edges) among image features (considered in the transformation space) upon which the detection is sure. Their main assumption is that the features are uniformly distributed. This can look odd, because images are precisely made of non-uniformly distributed features (the edgels are along edges ...). Let us quote Grimson and Huttenlocher's answer: *"Although the features in an image appear to be clustered (e.g. shadow edges and real edges), this does not necessarily mean that a uniform random distribution is a bad model. Even under a random process, clusters of events will occur. The question is whether the degree of clustering of features more than one would be expected from a random process."* The last sentence is exactly the formulation of an *a contrario* model, that we will soon define. This framework allows the authors to estimate the probability that a certain cluster is due to chance (due to the *"conspiracy of random"* in their words). Fixing a threshold on this probability gives sure detections: rare events are the most significant.

In the following, we intend to make such probabilistic methods reliable and giving the right matching thresholds.

1) Instead of defining a threshold distance for each query shape, we define a quantity (namely the Number of False Alarms) that can be thresholded independently of the query shape.

2) This quantity can be interpreted as the expected number of appearances of a query shape at a certain distance. Even if thresholding this number naturally leads to thresholding the matching distance, we get an additional information about how likely the matching is, and therefore about how sure we are that the matching is correct.

Anterior false alarm rates methods

What we propose is inspired by signal processing theory, in which the concept of controlling the detection quality by the "Number of False Alarms" was introduced. Let us be more precise. In [3], the authors are interested in detecting gravitational waves. They assume that the detector noise is white, stationary, and gaussian with zero mean and standard deviation σ. The problem consists in separating signal and noise. A signal $(x_i)_{1\leq i\leq N}$ of N data samples being given, they count the number of samples whose values exceed a threshold $s\cdot\sigma$. In the absence of signal, the noise being gaussian:

$$\Pr(|x_i|\geq s\sigma) = 2\int_s^{+\infty}\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx.$$

If N_c is the number of samples above threshold, one has:

$$\Pr(N_c = n) = \binom{N}{n}p^n(1-p)^{N-n},$$

where $p = \operatorname{erfc}(s/\sqrt{2})$, erfc is the complementary error function. Under limit considerations, if $\mu_c = Np$ and $\sigma_c = \sqrt{Np(1-p)}$, the normalized random variable $\tilde{N}_c = (N_c-\mu_c)/\sigma_c$ is well approximated by a standard normal random variable. The threshold s being set by physical arguments, the following relation between the detection threshold η (the number of samples upon which it is unlikely that the signal was processed by the noise) and the false alarm rate r_{fa} holds:

$$2\cdot\frac{1}{\sqrt{2\pi}}\int_\eta^{+\infty}\exp\left(\frac{-x^2}{2}\right)dx = r_{fa}.$$

The authors fix r_{fa} by converting it to the *number of false alarms* per hour for their sampling rate, and thus deduce a handy value for η. Of course, the lower the threshold, the higher the number of false alarms, and conversely.

To summarize, a detection threshold is deduced by the imposed value of the number of false alarms of the detection algorithm. This viewpoint is exactly the same as what we develop in the following.

Another example for image processing can be found in [8]. The aim of the authors is to detect small targets in natural images, which are modelled as texture images (the targets are supposed to be rare enough in order not to disturb this model). Thanks to a convenient transformation, the grey level distribution over the images is assumed to be a zero-mean unit-variance gaussian. Grey levels in small windows are then observed. Although there is no reason why the grey levels over a fixed small window should follow a gaussian distribution, the authors observe that the gaussian model is well fitted. In other words, they build a background model for these small windows. A rare window with regards to this model (*i.e.* whose probability is less than a certain threshold) is supposed to correspond to a detection. The detection threshold is fixed in such a way that the false alarms rate is low enough.

Another example of target detection in non-Gaussian images is given in [33]. The authors model the background of the considered images with a fractal model based on a wavelet analysis. Targets are detected as rare events with regard to this model.

A preliminary, less efficient of the method presented here was also proposed in [24, 23].

The plan of this article is as follows. In section 2, we introduce the notion of *meaningful match*. This concept allows to rank matches with a given curve by a criterion which is given an accurate and handy meaning: the Number of False Alarms (NFA). In fact, the NFA is intuitive enough to furnish a 'natural' detection threshold. For the sake of clarity, in section 3 we briefly describe the shape feature extraction and encoding algorithm for which we build such a decision rule. The background model for the shape database is described in section 4. Experimental results in section 5 show that the proposed decision rule fits indeed well the theory. In section 6 we explain that such a model must fit a highly desirable characteristic of recognition: it is relative to the database and to the query shape. Decision theory is often modelled through hypothesis testing; consequently we discuss in section 7 the way this model can be seen in that framework. Section 8 deals with some perspectives.

2 Meaningful matches and *a contrario* model

The aim of what follows is to remove any degree of arbitrariness in fixing the acceptation / rejection threshold of the decision 'such shape matches such other one'.

We shall first dress up an empirical statistical model of the shape database. The relevant matches will be detected *a contrario* as rare events for this model. This detection framework has been recently applied by Desolneux *et al.* to the detection of alignments [9] or contrasted edges [10], by Almansa *et al.* to the detection of vanishing points [1], and by Cao to the detection of good continuations [6]. The main advantage of this technique is that the only parameter that controls the detection is the Number of False Alarms, a quantity that measures the rareness of an event, and that has a handy meaning.

Suppose that the problem is to decide whether a shape code matches some other shape code from a database of N_B codes. The general settlement we consider only assumes that the shape codes are lists of n features, each of them belonging to a metric space (E_i, d_i), $1 \leq i \leq n$. Let $X = (x_1, \ldots, x_n)$ be a query shape code, and let $Y = (y_1, \ldots, y_n)$ denote an element of the cartesian product $E_1 \times \cdots \times E_n$. Given a positive real number δ, we say that X and Y are δ-close if and only if:

$$\forall i \in \{1, \ldots, n\},\ d_i(x_i, y_i) \leq \delta.$$

Two codes match if they are δ-close, with δ *small enough*: we have to set a threshold for δ. In this section, we assume that we can compute the probability (denoted by $\mathcal{P}(X,\ \delta)$ in what follows) that a shape code Y is δ-close to X. One way to do so is to define the features so that the n random variables $y \mapsto d_i(y, x_i)$ are mutually independent. Then one simply has

$$\mathcal{P}(X,\ \delta) := \Pr(Y \text{ s.t. } Y \text{ is } \delta\text{-close to} X) = \prod_{i=1}^{n} \Pr(y \in E_i \text{ s.t. } d_i(y, x_i) \leq \delta). \tag{1}$$

Each term of the former product can then be empirically estimated on the database: for each i, one computes the distribution function of $d_i(z, x_i)$, when z spans the i^{th} feature of the codes in the database.

Definition 1 (ε-meaningful match) *A shape code $X = (x_1, \ldots, x_n)$ being given, we say that another shape code $Y = (y_1, \ldots, y_n)$ matches X ε-meaningfully if:*

$$N_B \cdot \mathcal{P}(X,\ d) \leq \varepsilon,$$

where $d = \max_{i=1,\ldots,n} d_i(x_i, y_i)$.

Remark: Given $\varepsilon > 0$, one can compute the associated maximal distance $d^*(\varepsilon)$ such that $N_B \cdot \mathcal{P}(X, d) \leq \varepsilon$. This positive real number d^* is uniquely defined since the N functions $\delta \mapsto \Pr(y \in E_i \text{ s.t. } d_i(y, x_i) \leq \delta)$ (with $1 \leq i \leq n$) are non-decreasing, and so is their product; consequently, the function $\delta \mapsto \mathcal{P}(X,\ \delta)$ is pseudo-invertible. Each positive real d such that $d \leq d^*$ also satisfies $N_B \cdot \mathcal{P}(X,\ d) \leq \varepsilon$. Notice that the empirical probabilities take account of the 'rareness' or 'commonness' of a possible match; indeed the threshold d is less restrictive in the first case and more strict in the other. The following proposition shows that the number of ε-meaningful matches is controlled by ε. This provides a more intuitive way to control detections than just tuning the distance δ for each query.

Proposition 1 *The expectation of the number of ε-meaningful matches over the set of all shape codes in the database is less than ε.*

Proof: Let Y_j $(1 \leq j \leq N_B)$ denote the possible shape codes, and χ_j denote the indicator function of the event e_j: "Y_j matches ε-meaningfully X." Let $R = \sum_{j=1}^{N_B} \chi_j$ be the random variable representing the number of codes matching ε-meaningfully X. The expectation of R is $E(R) = \sum_{j=1}^{N_B} E(\chi_j)$. With definition 1, $E(R) = \sum_{j=1}^{N_B} \mathcal{P}(X,\ d)$, so $E(R) \leq \sum_{j=1}^{N_B} \varepsilon \cdot N_B^{-1}$, yielding $E(R) \leq \varepsilon$. ∎

The key point is that we control the expectation of R. Since dependencies between events e_j are unknown, we are not able to estimate the probability law of R. Nevertheless, the linearity still allows to compute the expectation. To end with these definitions, we provide a measure for the quality of matching.

Definition 2 (Number of False Alarms) *Given a shape code X and a distance $d > 0$, we call number of false alarms of a match with X at distance d the number:*

$$NFA(X,\ d) = N_B \cdot \mathcal{P}(X,\ d).$$

This framework provides a way to detect matches while controlling the number of objects that match purely by chance. If we want that 'random matches' between a query shape code and a database code appear only once on the average, we simply put $\varepsilon = 1$. If the query is made of N_Q equally relevant shape codes, and if we want to detect on the average at most one random match *over all query codes* (that is what we will do in section 5), we still put $\varepsilon = 1$ after replacing N_B by $N_B \cdot N_Q$ in definition 1 (in this case, proposition 1 obviously still holds).

3 From images to codes

Although this is not the aim of this article, we have to briefly explain the coding procedure of shapes we used. This method follows [19] and has no user parameter left. This means that it can be applied to any pair of images to perform a shape matching without any preliminary processing or parameter fixing.

3.1 Shape extraction and normalization

The described method has the three usual steps, namely extraction, filtering and encoding.

1. Consider the level lines in an image (*i.e.* the boundaries of the connected components of its level sets). This representation has several advantages. Although it is not invariant under *scene illumination* changes (in this case the image in itself is changed and no descriptor remains invariant), it is invariant under *contrast* changes. Moreover, the boundaries of the objects lying in the image are well represented by the union of some pieces of level lines. The mathematical morphology school has claimed that all shape information is contained in level lines, and this is certainly correct in that we can reconstruct the whole image from its level lines. Thus, a straightforward decision might be to define the shape elements of an image as the set of all of its level lines. Actually, this makes sense, since it is easily checked that significant parts of boundaries of objects seen in an image are contained in level lines. Thus, level lines can be viewed as concatenations of pieces of boundaries of objects of interest and therefore encode all shape information.

   Nevertheless, level lines provide redundant information (in some cases they are included in each other), or non-pertinent information (short lines in homogeneous areas of the image). This is why Desolneux *et al.* [10] (using a compact representation of level lines in images due to Monasse [21, 22]) developed the so called *maximal meaningful level lines* theory: this method computes all level lines that are long and contrasted enough to be relevant, and the model is parameter-free. The aim of this step is to reduce the number of shapes to be encoded, in order to speed up the decision stage. This step is parameterless.

2. Once the shapes are extracted, smooth them in order to eliminate aliasing effects (therefore the scale is such that the smoothing erases details on curves of size one pixel). The Geometric Affine Scale Space [28, 2] is fully convenient (since smoothing commutates with affine transforms):

$$\frac{\partial x}{\partial t} = |\mathrm{Curv}(x)|^{\frac{1}{3}} \vec{n}(x),$$

   where x is a point on a level line, $\mathrm{Curv}(x)$ the curvature and $\vec{n}(x)$ the normal to the curve, oriented towards concavity. We use a fast implementation by Moisan [20]. Once again, the aim is to reduce the amount of level lines in order to simplify the most pertinent ones, the final goal remaining the same: to make the decision stage faster. This step is parameterless, since the scale at which the smoothing is applied is fixed and given by the pixel size.

3. Build invariant representations (up to either similarity or affine transforms). Following [19], we define local frames for each level line, based on robust directions (tangent line at inflexion points or at flat points, or bitangent lines). Such a representation is obtained by uniformly sampling a piece of curve in this frame. Here is the procedure for similarity invariance (figure 1 a) illustrates it). Affine invariance is also theoretically possible and is explained in [18], but has not been yet investigated further at this point of our research. The semi-local encoding allows to deal with occlusion.

   In order to represent a level line $\mathcal{L}$, for each flat point, inflexion point, and for each couple of points on which the same straight line is tangent to the curve, do the following:

   a) Consider the tangent line $\mathcal{D}$ to this point or couple of points;

   b) Build the first two perpendicular lines $\mathcal{P}_1$ and $\mathcal{P}_2$ to $\mathcal{D}$ that are tangent to $\mathcal{L}$ (following a direction on the curve for $\mathcal{P}_1$, and the opposite for $\mathcal{P}_2$);

   c) Compute the coordinates of R_1 (resp. R_2), intersection point of $\mathcal{D}$ with $\mathcal{D}_1$ (resp. $\mathcal{D}_2$);

   d) Store the *normalized* coordinates of 45 equi-distributed points over an arc on $\mathcal{L}$ of length $5 \cdot \|R_1R_2\|$, centered at the intersection point of $\mathcal{L}$ with the perpendicular bissector of $[R_1R_2]$. "Normalized coordinates" mean coordinates in the similitude-invariant frame defined by the segment $[R_1R_2]$.

Let us remark that this method only allows to encode non-convex enough curves. Again, here we have a parameterless method. To be more precise, there are of course several numerical implementation parameters in the mentioned steps, like (e.g.) the mesh step for encoding the level lines. Now, all of these numerical parameters are fixed once for all experiments.

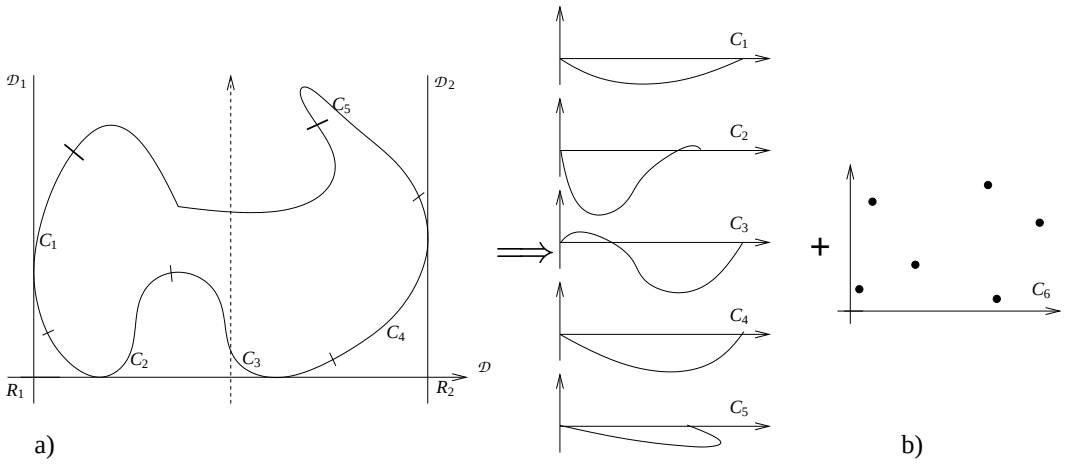


Figure 1: Sketch a): original shape in a normalized frame based on a bitangent line. Both ends of the piece of the shape of length $F \cdot \|R_1R_2\|$ are marked with bold lines: this representation is split into 5 pieces C_1, C_2, C_3, C_4, and C_5. Sketch b): each of them is normalized, and a sixth feature C_6 made of the endpoints of these pieces is also built.

3.2 Modelling

We now explain how a normalized representation can be processed in order to fit the framework of section 2. Following what we wrote, we have to deduce for each representation a N-tuple of statistically independent features. Many possibilities have been investigated, each of them coming up against the trade-off between independence of the descriptors and completeness of the description. To describe a shape accurately without redundancy is in fact tricky. We first thought of using the highest Principal Component Analysis components (though being not independent, PCA components are at least decorrelated). This gave good results, but limited to the fact that these components do not provide a complete description (see [24] for more details). Moreover, the number of components that are kept is touchy. In addition, the linear model is not really adapted to shapes.

The shape ressemblance and decision process needs the three following requirements.

a) A shape $\mathcal{F}$ looks like another shape $\mathcal{F}'$ up to an allowed deformation A if the Hausdorff distance between $\mathcal{F}$ and $A \cdot \mathcal{F}'$ is small enough.

b) The codes are L^∞-complete: two similar (for the L^∞ distance) sets of features should come from two alike shapes (up to the allowed deformations).

c) Independence requirement (equation 1): the functions $\mathcal{F}' \mapsto d_i(x_i(\mathcal{F}), x_i(\mathcal{F}'))$ are mutually independent.

Clearly, the PCA method does not fit any of the preceding requirements, except possibly c).

3.3 Six close to independent and complete shape features

The best procedure we found to achieve simultaneously a), b) and c), is the following (see figure 1 b) for an illustration). Each normalized representation C is split into five pieces of equal length. Now, each of these pieces is normalized by mapping the chord between its first and last points on the horizontal axis, the first point being at the origin: the resulting 'normalized small pieces of curve' are five features C_1, C_2, ... ,C_5. These features ought to be independent; nevertheless, C_1, ... ,C_5 being given, it is impossible to reconstruct the shape they come from. For the sake of completeness a sixth, global, feature C_6 is therefore made of the endpoints of the

five previous pieces, in the normalized frame. For each piece of level line, the 'code' introduced in section 2 is made of these six 'generic' shape features $C_1, \ldots, C_6$. The distances d_i between them are L^∞-distances.

We have in the database $\mathcal{B}$ many shape examples, which we consider as a good sampling of a background shapes model. We assume that the functions $\mathcal{F}' \mapsto d_i(x_i(\mathcal{F}), x_i(\mathcal{F}'))$ are mutually independent. A discussion of the mutual independence of these features will be led in section 4.

Now, why $5+1$ features ? Equation 1 shows a product of n probabilities (here, $n = 6$). Each of them is learnt upon the database, with the same accuracy. Consequently, the bigger is n, the lower values $P(X,\ d)$ can reach. Our aim is precisely to reach very low NFA, so that such shape is ensured to match such other. Nevertheless, n cannot be as large as we want, because finding features that make a complete describtion of a shape but are mutually independent is a hard problem. The $5+1$ features method appears as a convenient trade-off.

4 The background model

4.1 The background model and a detection terminology

Our aim is to decide between two possibilities: one is that two shapes match and the other one that the match has been produced casually. We decide for the first when the second is unlikely. So we need a model for the casual production of a shape $\mathcal{F}'$ close to a fixed shape $\mathcal{F}$: this is what we call the background model. We need a model because we cannot just rely on the empirical observation of a frequency: the evaluated probabilities will usually be too low to be observed in our database.

The background model is assumed to evaluate accurately the probability that a shape $\mathcal{F}'$ falls at a distance less than some δ from a given shape $\mathcal{F}$, namely $P(d(\mathcal{F}, \mathcal{F}') < \delta)$. According to the preceding discussion, the background model can be summarized in the following assumption:

Background model: given $\mathcal{F}$, the distance functions $\mathcal{F}' \mapsto d_i(x_i(\mathcal{F}), x_i(\mathcal{F}'))$ (for $i \in \{1, \ldots, n\}$) are mutually independent.

How good is this assumption will be tested. If the model is true, it will evaluate correctly the probability that $\mathcal{F}$ matches $\mathcal{F}'$ 'just by chance'. This leads us to the following terminology. Assume that two shapes $\mathcal{F}$ and $\mathcal{F}'$ fall at a distance δ. The smaller $NFA(\mathcal{F}, \delta)$ is, the more certain we are that $\mathcal{F}'$ cannot look like $\mathcal{F}$ just by chance. In the following, we call *ε-detection*, or simply *detection*, any $\mathcal{F}'$ such that $NFA(\mathcal{F}, d(\mathcal{F}, \mathcal{F}')) \leq \varepsilon$.

Although thresholding $NFA(\mathcal{F}, \delta)$ corresponds to a threshold of the distance δ, the Number of False Alarms is much more handy. Indeed, a uniform bound over $\mathcal{F}$ for $NFA(\mathcal{F}, \delta)$ can be imposed, leading automatically to a distance threshold $\delta^*(\mathcal{F})$ adapted to the considered code $\mathcal{F}$.

4.2 Testing the background model

The computation of the probability $\mathcal{P}(\mathcal{F},\ \delta)$ that a shape code match could be at a distance lower than δ to a (target) code $\mathcal{F}$ is correct under the independence assumption on the pieces of codes (formula 1). Of course, the degree of trust that we are able to give to the associated Number of False Alarms $NFA(\mathcal{F},\ \delta)$ (definition 2) strongly depends on the validity of this independence assumption. Before applying this methodology to realistic applications, we have to test the independence of the pieces of codes, and so to ensure the correctness of the methodology. We show in what follows that the pieces of codes obtained by the proposed normalization (section 3) are *not* independent. Nevertheless, experiments point out that detection under *a contrario* principle (*i.e.* a meaningful match is a match that is not likely to occur in a noise image) is fully satisfying.

4.2.1 Independence testing

In order to compute the probability $\mathcal{P}(X,\ \delta)$, the mutual independence of the 'pieces of codes' is needed. More precisely speaking, a code made of pieces x_i being given, the binary random variables $y \mapsto d_i(y, x_i) \leq \delta$ are supposed mutually independent.

We cannot estimate the joint probability

$$\Pr\left((y_1, \ldots, y_n) \in E_1 \times \cdots \times E_n \text{ s.t. } d_1(y_1, x_1) \leq \delta, \ldots, d_n(y_n, x_n) \leq \delta\right)$$

(estimating the law of this random vector would indeed require too many samples) and compare it to the product of the probabilities $\prod_{i=1}^{n} \Pr(y_i \in E_i \text{ s.t. } d_i(y, x_i) \leq \delta$. On the other hand, it turns out that the joint

probability associated to two pieces of codes can be accurately enough estimated. Thus, instead of testing the mutual independence of the pieces of codes, we merely test the independence pairwise.

Let us explain the Chi-square test framework, applied to independence testing [26]. Two binary random variables X and Y being given, let us denote $p_{ij} = P((X,Y) = (i,j))$ for i and j in $\{0,1\}$: these probabilities are empirically estimated over samples following the laws of X and Y. Thus, $P(X = i) = p_{i0} + p_{i1}$ and $P(Y = j) = p_{0j} + p_{1j}$. If independence assumption holds, we have: $p_{ij} = P(X = i) \cdot P(Y = j)$. The Chi-square statistical test consists in evaluating the difference between the expected number of samples such that $(X,Y) = (i,j)$ if this assumption were true, and the observed number of samples. If N is the number of samples following the law of (X,Y), and if O_{ij} is the observed number of samples (i,j), let us compute:

$$\chi^2 = \frac{(N(p_{00}+p_{01})(p_{00}+p_{10}) - O_{00})^2}{N(p_{00}+p_{01})(p_{00}+p_{10})} + \frac{(N(p_{10}+p_{11})(p_{10}+p_{00}) - O_{10})^2}{N(p_{10}+p_{11})(p_{10}+p_{00})} + \frac{(N(p_{01}+p_{00})(p_{01}+p_{11}) - O_{01})^2}{N(p_{01}+p_{00})(p_{01}+p_{11})} + \frac{(N(p_{11}+p_{10})(p_{11}+p_{01}) - O_{11})^2}{N(p_{11}+p_{10})(p_{11}+p_{01})}.$$

This quantity can be assumed to follow a Chi-square distribution with one degree of freedom, if enough samples are provided in order to estimate accurately the probabilities p_{ij}, and the O_{ij}.

Of course, the lower is χ^2, the likelier we are to accept the hypothesis, and vice-versa. By comparing the obtained value with the quantiles of the Chi-square law, we are able to accept or reject the hypothesis (independence between the random variables), with a certain significance level.

We have led this experiment with the binary random variables associated to the codes that we introduced in what precedes, with different target codes and database codes. The results are clear: we are able to reject the independence assumption with a high significance level. Nevertheless, the rejection is strong because the tested databases are very large: Chi-square test is all the more accurate (and so is the rejection confidence) as the number of samples is large. In other terms, a "slight" dependence with a large number of samples leads to a very significant rejection ; this means that the Chi-square test does not yield an absolute measurement of how dependent or how independent variables are.

The next section shows that the independence assumption is true enough to keep the *a contrario* detection principle true, in a sense that will be made clear.

4.2.2 Number of detections testing

The purpose of this experiment is to test the main properties of the proposed method, namely the control of the expected number of 'random' detections (the Number of False Alarms, proposition 1). Number of False Alarms computation holds under the independence assumption. If this assumption is true, and if the database contains no copy of a sought code, the Number of Detections of this code in the database with a NFA lower than ε should be lower than ε. If this is not the case, then we can ensure that the hypothesis are violated: either the independence assumption is false, or there is some actual matches with the sought code in the database. The Chi-square test proved that, strictly speaking, the independence assumption is not valid. Nevertheless, the following experiments show that in a random situation (no other causality between codes than to be realization of the same random process), the NFA is a pretty good prediction of the number of detections. The independence assumption is enough valid, so that proposition 1 still holds.

As a first experiment we check the detection thresholds on a very simple model: we consider as code database and code query some random walks with independent increments. In this case the background model is ensured to be true, in sense that the considered codes fit perfectly the independence assumption.

Table 1 shows that the Number of False Alarms is very accurately predicted for various database sizes: the number of detections (which are pure 'random' detections) with a NFA lower than ε is about ε indeed.

100,000 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0	0	2.3	15.2	122.2	1,075.5	9,872.2
50,000 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0.2	0.3	1.5	11.9	106.1	1,001.1	9,789.5
10,000 codes, value of ε:	0.01	0.1	1	10	100	1,000	
Numb. of det. with $NFA < \varepsilon$:	0	0	1.2	12.5	108.4	985.0	

Table 1: Random walks. Average (over 10 samples) number of detections vs ε. Tabular 1: database of 100,000 codes. Tabular 2: database of 50,000 codes. Tabular 3: database of 10,000 codes.

Of course, modelling codes with random walks is not realistic. As proved in what precedes, distances between codes are actually *not* independent. In our opinion, the lack of independence comes from two points. On the one hand, codes correspond to pieces of level lines, and consequently they are constrained to not self-intersect. On the other hand, codes are normalized, and show therefore structural similarities (for example, codes coming from bitangent points show mostly common structures). In order to quantify the 'amount of dependency' due to these two points, we have led the two following experiments.

Table 2 shows the number of detections *versus* the number of false alarms for databases made of pieces of level lines (*not* normalized, the codes are just made out of 45 consecutive points on pieces of level lines). Consequently, the obtained codes are constrained not to self-intersect. In this experiment, the independence is only spoiled by this property, not by the normalization. Although the Chi-square test shows that the codes are not independent, once again the number of detections is accurately predicted: the number of matches with a NFA less than ε is indeed about ε.

101,438 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0.1	0.1	1.7	13.8	95.3	942.5	9,789.4

50,681 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0	0	1.2	10.3	90.5	955.1	9,859.3

9,853 codes, value of ε:	0.01	0.1	1	10	100	1,000
Numb. of det. with $NFA < \varepsilon$	0	0.1	0.9	9.5	94.3	973.1

Table 2: Pieces of white noise level lines. Average (over 10 samples) number of detections vs ε. Tabular 1: database of 101,438 codes. Tabular 2: database of 50,681 codes. Tabular 3: database of 9,853 codes.

As a last experiment, let us consider databases made of normalized codes extracted from pieces of level lines in white noise images. Table 3 shows that the number of detections is still of the same magnitude as the number of false alarms ε, but is not as precisely predicted as in the latest experiments. Roughly speaking, it means that 'most of the dependence' comes from the normalization procedure, and not from the non-self-intersection constraint. Nevertheless, the order of magnitude is still correct, and does not depend on the size of the database. These properties are enough in order to set the Number of False Alarms threshold under the *a contrario* methodology. Following this method, a match is supposed to be highly relevant if it cannot happen in white noise images. According to table 3, matches with a NFA lower than 0.1 are ensured to be impossible in white noise images. The following realistic experiments (section 5) will thus be led with a detection threshold equal to or less than 0.1.

104,722 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000	100,000
Numb. of det. with $NFA < \varepsilon$:	0.3	1.5	6.5	31.5	173.9	1,264.4	9,803.1	99,899.5

47,033 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0.1	0.3	3.7	20.2	125.4	976.3	9,854.2

10,784 codes, value of ε:	0.01	0.1	1	10	100	1,000
Numb. of det. with $NFA < \varepsilon$:	0	0.2	2.6	14.8	107.6	973.3

Table 3: Normalized pieces of white noise level lines. Average (over 10 samples) number of detections vs ε. Tabular 1: database of 104,722 codes. Tabular 2: database of 47,033 codes. Tabular 3: database of 10,784 codes.

Table 4 shows the number of detections *versus* number of false alarms for a database made of normalized long (length greater than 135 pixels) pieces of level lines from white noise images. The results are not better than in the preceding experiment, we cannot assert that the independence violation is due to short pieces of level lines.

101,743 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000	100,000
Numb. of det. with $NFA < \varepsilon$:	0	0.4	2.8	18.5	124.3	1,123.2	9,693.8	99,921.0

51,785 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0	0.3	2.9	16.0	118.6	983.4	9,800.4

11,837 codes, value of ε:	0.01	0.1	1	10	100	1,000	10,000
Numb. of det. with $NFA < \varepsilon$:	0	0.2	1.4	12.3	105.9	975.2	9,974.7

Table 4: Normalized long pieces of white noise level lines. Average (over 10 samples) number of detections vs ε. Tabular 1: database of 101,743 codes. Tabular 2: database of 51,785 codes. Tabular 3: database of 11,387 codes.

5 Experiments

Though images and pieces of level lines superimposed to images are shown, the reader should keep in mind that the decision rule actually only deals with *normalized codes*. However, the results for pieces of level lines are shown for the sake of clarity.

5.1 Casablanca

The following experiment consists in comparing the codes extracted from two different posters of the movie *Casablanca* (see figure 2). Many slight differences can be seen, such as modification of the painting of the actors' face, or the width of the letters making the title. We consider as query codes the codes extracted from image 1 (4,645 codes) and as database codes the codes extracted from image 2 (11,792 codes). Using section 2 notations, $N_Q = 4,645$, and $N_B = 11,792$. The level lines are extracted from images with the algorithm described in [7].

Figure 2: Casablanca original images (on the left) and level lines (on the right). Top: image 1 (searched level lines). Bottom: image 2 (database level lines).

Figure 3 shows on the left the set of pieces of level lines in image 1 that match a piece of level line in image 2 with a corresponding Number of False Alarms less than 1 (first row) and less than 10^{-1} (second row), and on the right the set of codes of image 2 that correspond to at least one code in image 1. A total number of 261 matches are identified. As predicted by the theory, no false match exhibits a NFA lower than 10^{-1}.

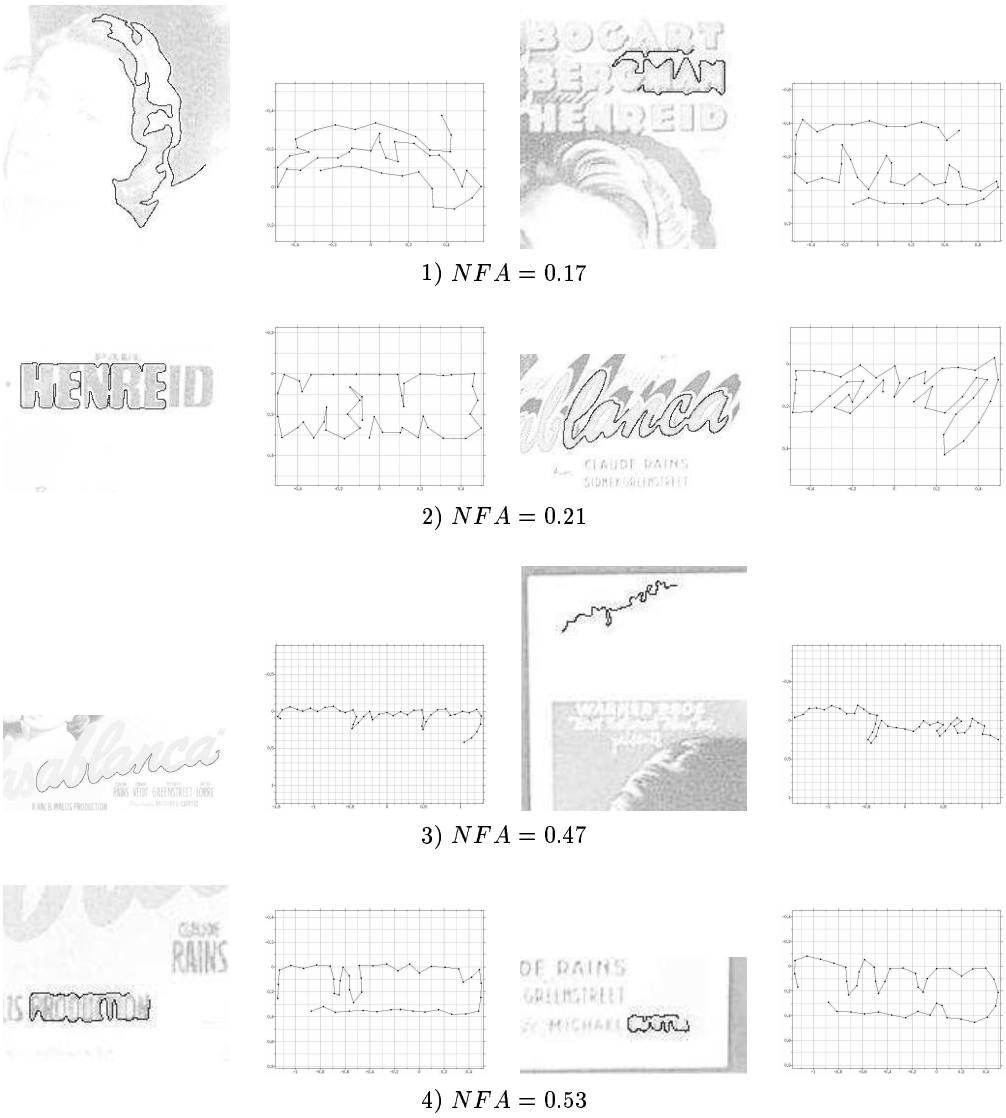
We have claimed that our Number of False Alarms between two codes is a good quality estimation. The 'best' match exhibits a NFA of 5.10^{-13} (see the two corresponding curves at figure 4). It means that if the query image is compared to $1/(5.10^{-13})$ images, then there should be at most one shape matching with the query shape of figure 4. Such a situation is predicted by our theory, since both curves are at the same time long, complicated and correspond very well.

Figure 5 shows the four wrong matches. All wrong matches have a NFA between 1 and 0.1. We could expect such a NFA since the curves are long, show local variations, but their global aspect is the same, as it can be seen on the corresponding codes.

Figure 3: Casablanca matches. To each bold piece of level line on the right corresponds a bold level lines on the left. Top: $NFA < 1$. Bottom: $NFA < 10^-1$, no false match can be seen.

$NFA = 5 \cdot 10^{-13}$

Figure 4: The match with the lowest NFA. The query shape (left-hand) matches the database shape (right-hand).



1) $NFA = 0.17$

2) $NFA = 0.21$

3) $NFA = 0.47$

4) $NFA = 0.53$

Figure 5: The four false matches if the NFA threshold is set to 1, with their codes. On the left: query shape and code; on the right: base shape and code. The corresponding normalized representation are roughly similar but show local variations.

5.2 Weather

We look for level lines extracted from two successive frames a satellite animation (see figure 6).

Number of codes in the request image: 5,002.

Number of codes in the database image: 6,071.

Results can be seen on figure 7. A total number of 535 matches are identified when the NFA threshold is set to 1

Only three false matches (among matches with a NFA lower than 1) appear : see figure 8.

We can see that the quality of the matches (measured by NFA) is the poorest for shapes belonging to the 'cyclonic structure' that presents strong variations between the images, and is the highest for the piece of the images that does not change: the French Brittany and Normandy coast (with a NFA of $3 \cdot 10^{-11}$) (figure 9).

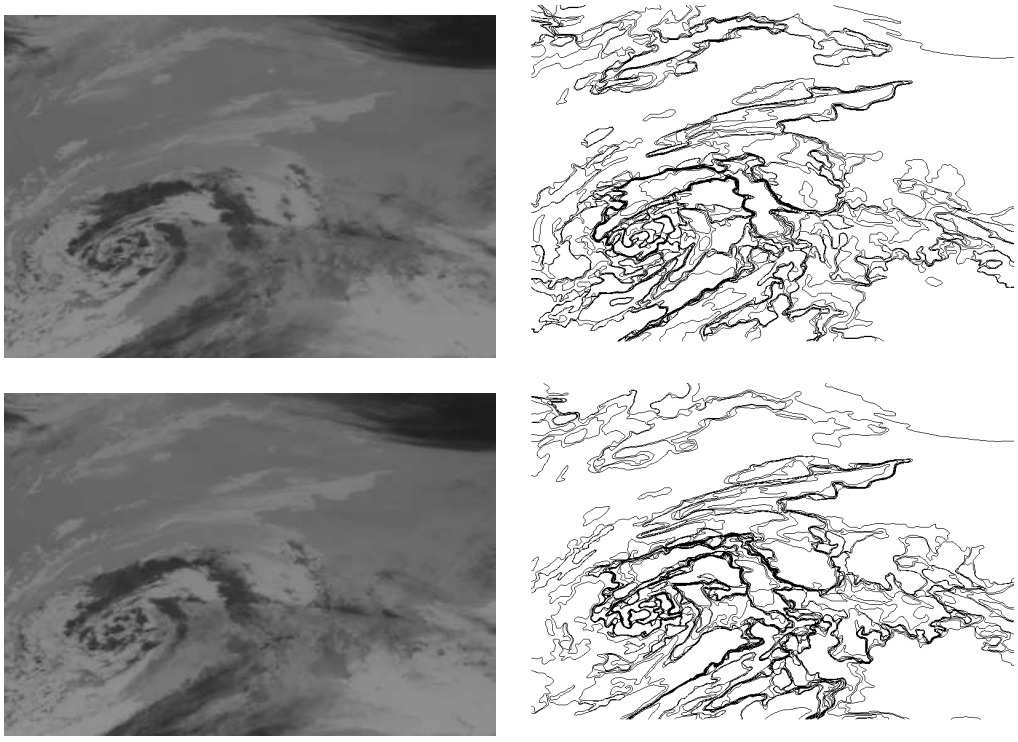
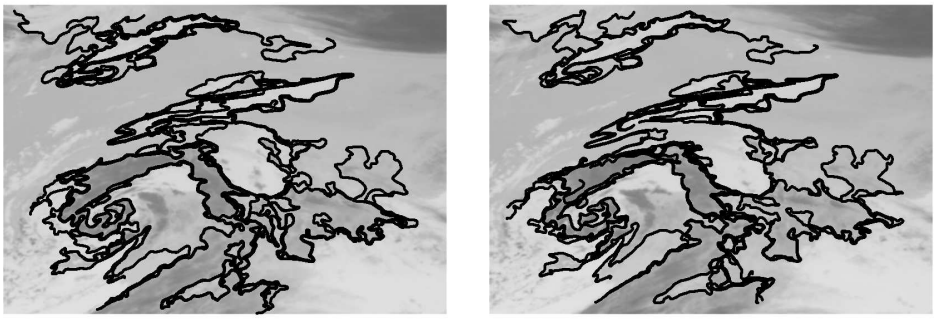
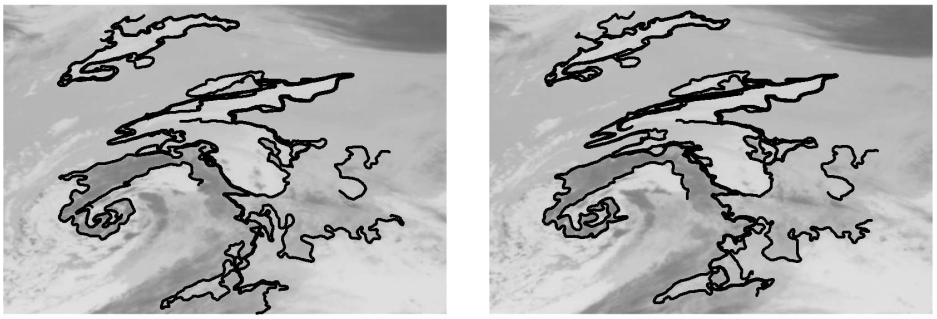


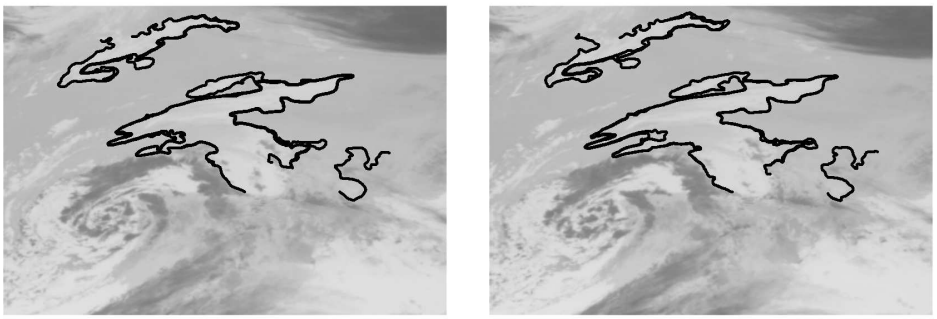
Figure 6: Weather original images (on the left) and level lines (on the right). Top: image 1 (searched level lines). Bottom: image 2 (database level lines).



1) $NFA \leq 1$

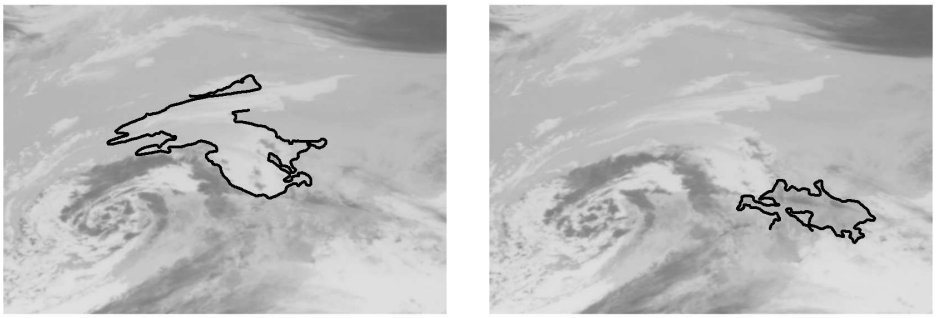


2) $NFA \leq 10^{-2}$

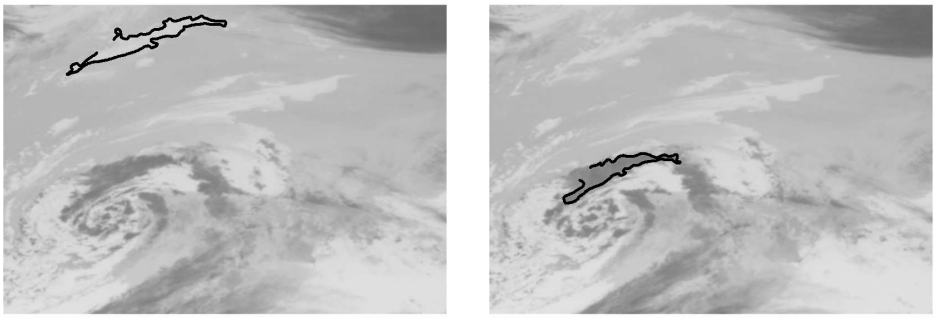


3) $NFA \leq 10^{-4}$

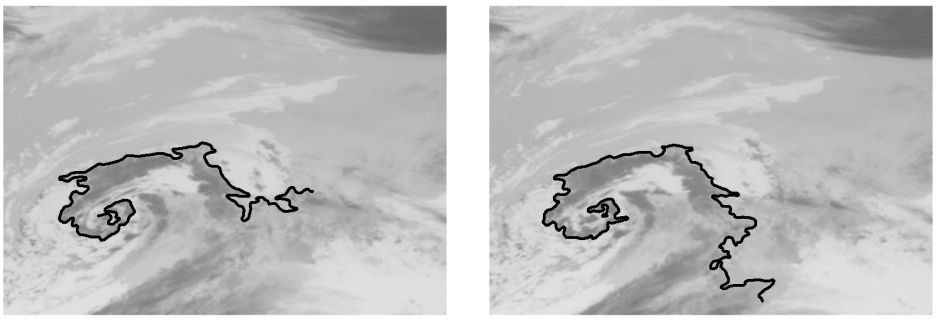
Figure 7: Weather: results. On the left: matches from the query. On the right: matches from the database. We can see that the 'worst' matches (belonging to the disturbed area) exhibit the highest NFA.



1) $NFA = 0.176$



2) $NFA = 0.786$



3) $NFA = 0.389$

Figure 8: The only three weather false matches. As expected, their NFA is between 1 and 10^{-1}.

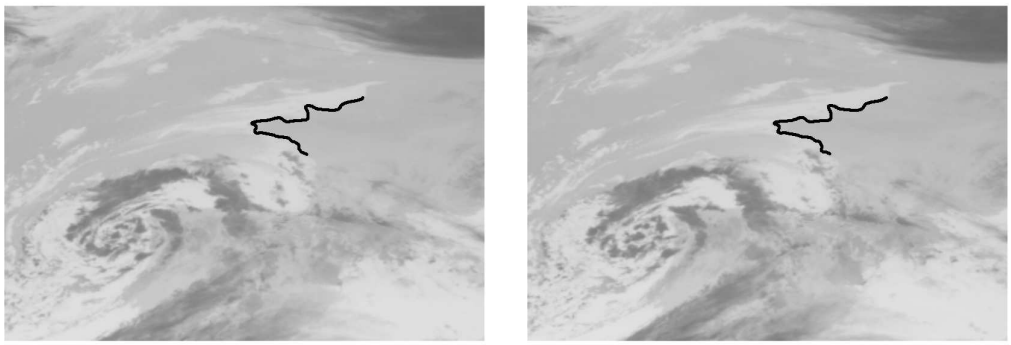


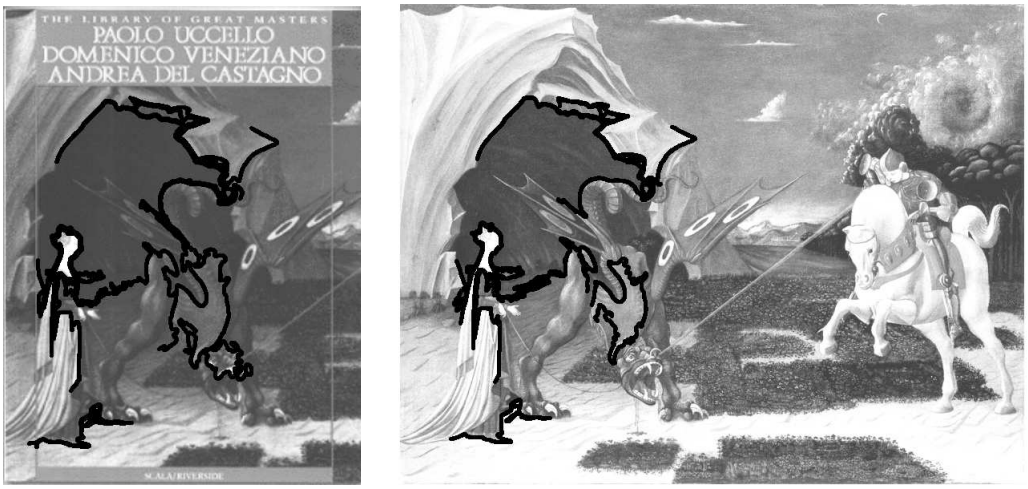
Figure 9: Weather best match: French Brittany and Normandy coast ($NFA = 3 \cdot 10^{-11}$).

5.3 Uccello

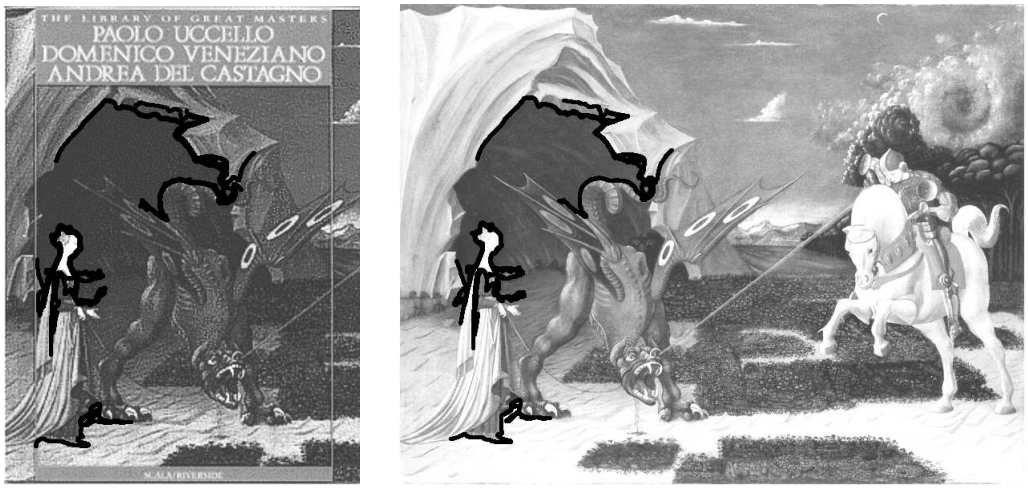
This experiment deals with searching the image at the top of figure 10 into the image at the bottom (two representations of *Saint George and the Dragon* by Uccello). Figure 11 shows detections for decreasing values of NFA thresholds.

As we can expect, if the NFA threshold is less than 10^{-1}, no false detection appears.

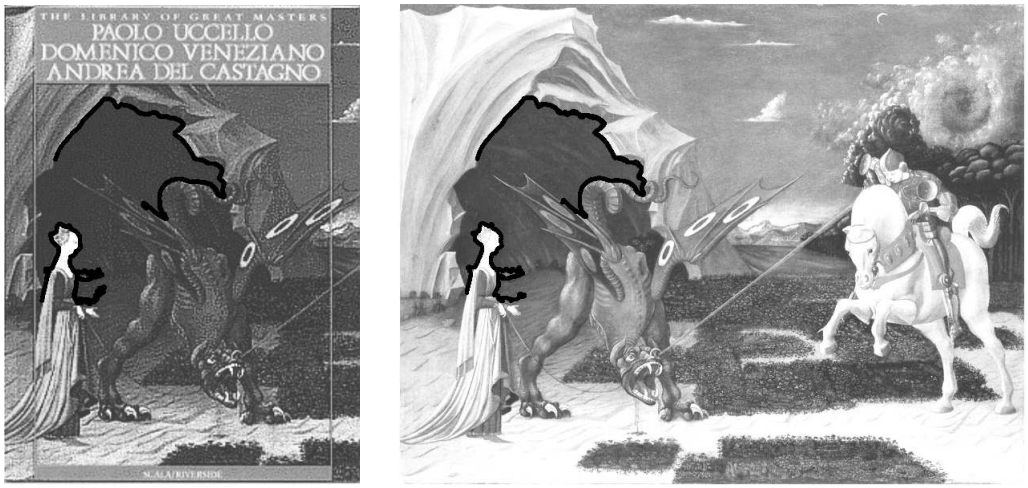
Figure 10: Uccello original images (on the left) and level lines (on the right). Top: image 1 (searched level lines, corresponding to $5,015$ codes). Bottom: image 2 (database level lines, corresponding to $11,243$ codes).



1) $NFA \leq 10^{-1}$



2) $NFA \leq 10^{-2}$



3) $NFA \leq 10^{-4}$

Figure 11: Uccello matches. No wrong match can be seen when the NFA threshold is set to 10^{-1}. Many good matches show a NFA lower than 10^{-4}: the lower the NFA, the higher the confidence.

6 Recognition is relative to the database

This experiment shows that the recognition threshold provided by the proposed algorithm depends on the database, and therefore on the context. We search for codes from the character 'm' (4 codes) (see figure 12) into 14 scanned page. We lead two experiments: in the first one the database that is used to learn probabilities is made of these 14 scanned pages (79,376 codes), whereas in the second one the database is made of 21 'natural' images (98,857 codes).

In order to get enough codes from the 'm', we use the "3+1 pieces method", each code being made of 27 points.

Figure 13 shows the codes from the scanned page in the background that match with some codes of the query, when the learning is processed with the scanned text (we can notice that all 'm' are recognized). Figure 14 shows the recognition result when the scanned text database is replaced by the natural image database. We can see that the recognition thresholds are more permissive in the second case (figure 15). This result is fully coherent with the theory: in the first case, the focus is put on recognition of shapes that share a common structure with 'm' *among other letters*, that is to say other 'm', whereas in the second case, we are interested in recognition of shapes that share a common structure with 'm' *among all natural shapes we can meet*, that is to say other 'similar' letters (that is why we get italic 'm' and other bad matches).

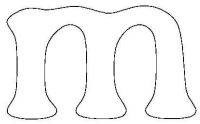
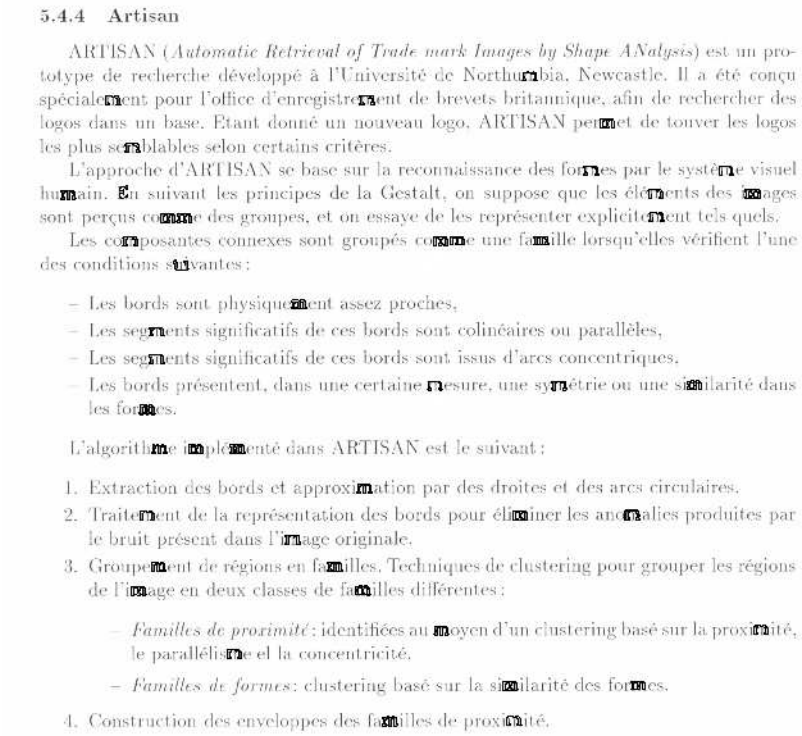


Figure 12: Characters - the query curve.



5.4.4 Artisan

ARTISAN (*Automatic Retrieval of Trade mark Images by Shape ANalysis*) est un prototype de recherche développé à l'Université de Northumbia, Newcastle. Il a été conçu spécialement pour l'office d'enregistrement de brevets britannique, afin de rechercher des logos dans un base. Etant donné un nouveau logo, ARTISAN permet de touver les logos les plus semblables selon certains critères.

L'approche d'ARTISAN se base sur la reconnaissance des formes par le système visuel humain. En suivant les principes de la Gestalt, on suppose que les éléments des images sont perçus comme des groupes, et on essaye de les représenter explicitement tels quels.

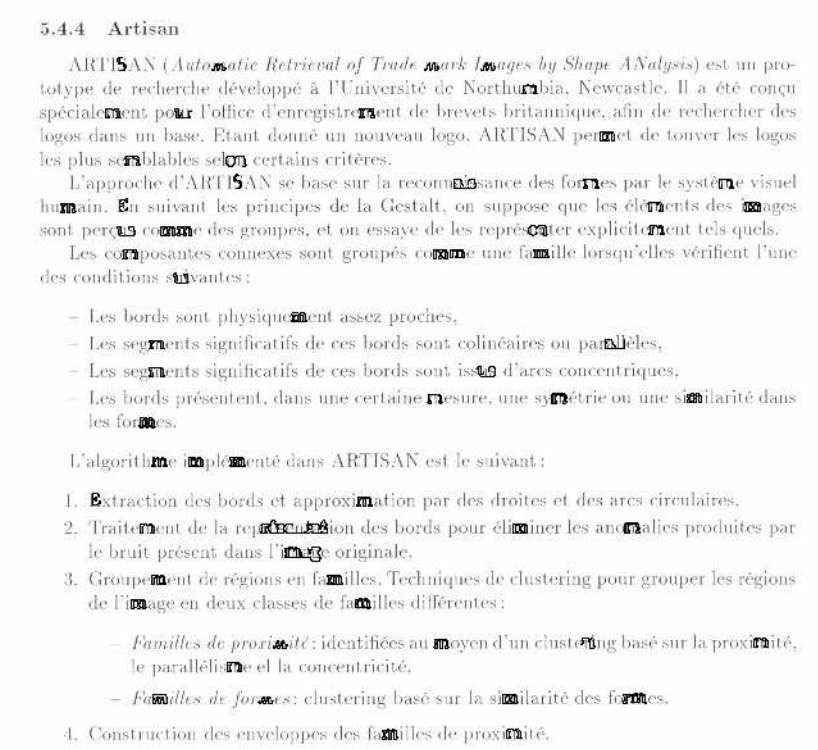
Les composantes connexes sont groupés comme une famille lorsqu'elles vérifient l'une des conditions suivantes:

- Les bords sont physiquement assez proches,
- Les segments significatifs de ces bords sont colinéaires ou parallèles,
- Les segments significatifs de ces bords sont issus d'arcs concentriques,
- Les bords présentent, dans une certaine mesure, une symétrie ou une similarité dans les formes.

L'algorithme implémenté dans ARTISAN est le suivant:

1. Extraction des bords et approximation par des droites et des arcs circulaires.
2. Traitement de la représentation des bords pour éliminer les anomalies produites par le bruit présent dans l'image originale.
3. Groupement de régions en familles. Techniques de clustering pour grouper les régions de l'image en deux classes de familles différentes:
   - *Familles de proximité*: identifiées au moyen d'un clustering basé sur la proximité, le parallélisme el la concentricité.
   - *Familles de formes*: clustering basé sur la similarité des formes.
4. Construction des enveloppes des familles de proximité.

Figure 13: Characters - Recognition with scanned text database. 111 matches.



5.4.4 Artisan

ARTISAN (*Automatic Retrieval of Trade mark Images by Shape ANalysis*) est un prototype de recherche développé à l'Université de Northumbia, Newcastle. Il a été conçu spécialement pour l'office d'enregistrement de brevets britannique, afin de rechercher des logos dans un base. Etant donné un nouveau logo, ARTISAN permet de touver les logos les plus semblables selon certains critères.

L'approche d'ARTISAN se base sur la reconnaissance des formes par le système visuel humain. En suivant les principes de la Gestalt, on suppose que les éléments des images sont perçus comme des groupes, et on essaye de les représenter explicitement tels quels.

Les composantes connexes sont groupés comme une famille lorsqu'elles vérifient l'une des conditions suivantes:

- Les bords sont physiquement assez proches,
- Les segments significatifs de ces bords sont colinéaires ou parallèles,
- Les segments significatifs de ces bords sont issus d'arcs concentriques,
- Les bords présentent, dans une certaine mesure, une symétrie ou une similarité dans les formes.

L'algorithme implémenté dans ARTISAN est le suivant:

1. Extraction des bords et approximation par des droites et des arcs circulaires.
2. Traitement de la représentation des bords pour éliminer les anomalies produites par le bruit présent dans l'image originale.
3. Groupement de régions en familles. Techniques de clustering pour grouper les régions de l'image en deux classes de familles différentes:
   - *Familles de proximité*: identifiées au moyen d'un clustering basé sur la proximité, le parallélisme el la concentricité.
   - *Familles de formes*: clustering basé sur la similarité des formes.
4. Construction des enveloppes des familles de proximité.

Figure 14: Characters - Recognition with natural images database. 154 matches.

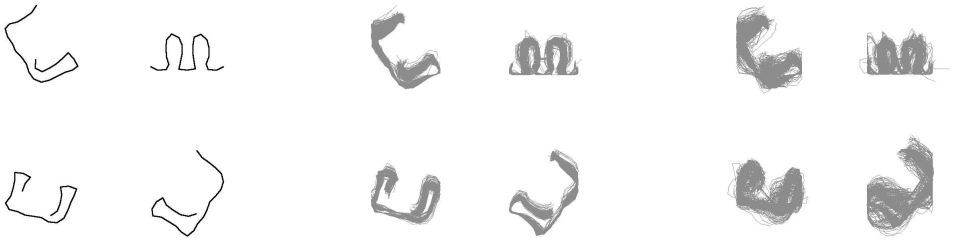


Figure 15: Characters - Superimposition of the matching codes. Left: query codes. Middle: superimposed matching codes with scanned text learning base. Right: superimposed matching codes with natural images learning base. We can see that the matching threshold is higher in the latter case.

7 Discussion about hypothesis testing

Hypothesis testing (see [11, 30]) is the branch of statistical inference that gives a formalism to decision theory. As far as we are interested in making binary decision (acceptation / rejection of an hypothesis), a statistical test (*i.e.* a decision rule) is equivalent to building up a partition of the set of observations Ω between two disjoint sets Ω_0 and Ω_1, the former corresponding to the observations that are consistent with the considered hypothesis $\mathcal{H}_0$, the latter to the observations that are not consistent with this hypothesis and that are rejected (we denote by $\mathcal{H}_1$ this counter-hypothesis in the following). Some authors call also $\mathcal{H}_0$ "null hypothesis," and $\mathcal{H}_1$ "alternative hypothesis."

The quality of a statistical test is measured by the probability of taking wrong decisions. Two kinds of error are possible: reject an observation ω though it is true (type I error, mis-detection), and accept ω though it is false (type II error, false positive). With each type of error we can associate a probability.

If, for an observation ω, $\mathcal{L}(\omega|\mathcal{H}_0)$ and $\mathcal{L}(\omega|\mathcal{H}_1)$ are the respective likelihood under the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, let:

$$\alpha = \int_{\Omega_0} \mathcal{L}(\omega|\mathcal{H}_1)d\omega\,,$$

$$\alpha_2 = \int_{\Omega_1} \mathcal{L}(\omega|\mathcal{H}_0)d\omega\,,$$

and:

$$\beta = 1 - \alpha_2 = \int_{\Omega_0} \mathcal{L}(\omega|\mathcal{H}_0)d\omega\,;$$

α (associated with type I error) is the *probability of false alarm*, and β is the *power function* of the test (α_2 is the *probability of non-detection* or *probability of a miss*, associated with type II error).

It is clear that the lower is α and the higher is β, the better is the test, but it is also clear that α and β cannot be optimized independently. Classically, ROC curves (Receiver Operating Characteristic curves) representing $\beta = f(\alpha)$ are associated with a statistical test $\mathcal{T}$. Robust tests show characteristic ROC curves looking like a Heaviside step: if α is near to 0, β should be near to 1.

The problem is to model the trade-off between α and β (or equivalently α_2). Let us examine two standard possibilities.

1. **Likelihood ratio test**. If we look for powerful tests (*i.e.* tests with the lowest rate of non-detection) among the tests which probability of false alarm α is bounded (by α^*), Neyman-Pearson lemma ensures that the most powerful test is the following likelihood ratio test:

   Classify the observation ω in Ω_0 if $\dfrac{\mathcal{L}(\omega|\mathcal{H}_1)}{\mathcal{L}(\omega|\mathcal{H}_0)} < h$ and in Ω_1 otherwise, where h is solution of :

   $$\int_{\left\{\omega\in\Omega,\,\frac{\mathcal{L}(\omega|\mathcal{H}_1)}{\mathcal{L}(\omega|\mathcal{H}_0)}\geq h\right\}} \mathcal{L}(\omega|\mathcal{H}_0)d\omega = \alpha^*.$$

   Knowing the two likelihood functions is necessary in order to achieve this test. Moreover, the value of α^* has to be fixed, what can be tricky.

2. **Bayesian test**. A test $\mathcal{T}$ being given, it is also possible to model the trade-off between α and α_2 by a weighted sum (Bayes cost): $J(\mathcal{T}) = p_0\alpha + p_1\alpha_2$, where p_0 (resp. p_1) is the prior probability of hypothesis $\mathcal{H}_0$ (resp. of counter-hypothesis $\mathcal{H}_1$) (p_0 and p_1 verify $p_0 + p_1 = 1$). It can be shown that the classification test that minimizes J is:

   Classify the observation ω in Ω_0 if $\mathcal{L}(\omega|\mathcal{H}_0)\cdot p_0 > \mathcal{L}(\omega|\mathcal{H}_1)\cdot p_1$ and in Ω_1 otherwise.

   Not only is the likelihood functions necessary, but also the prior probability of the hypothesis. Compared with the likelihood ratio test, there is no need of an arbitrary threshold.

In fact, if we write the Bayesian test such that:

$$\frac{\mathcal{L}(\omega|\mathcal{H}_1)}{\mathcal{L}(\omega|\mathcal{H}_0)} < \frac{p_0}{p_1}\,,$$

we realize that the ratio $\frac{p_0}{p_1}$ plays essentially the same role as a the parameter $h(\alpha^*)$ in the Neyman-Pearson theory. 'Bayesians' such as Jaynes [15] argue that each test can be explained as a bayesian test somehow or other. Let us quote Grenander [13]: *"Suffice is to say that when the notion of a prior makes sense and when there is sufficient knowledge about this prior we cannot afford to throw away this subject matter information: a Bayesian treatment is called for."*

This theoretical framework has obvious practical limits. Assuming the knowledge of the likelihood of both hypothesis ($\mathcal{L}(\omega|\mathcal{H}_0)$) and counter-hypothesis ($\mathcal{L}(\omega|\mathcal{H}_1)$) is not very realistic. If we want that the likelihood of the hypothesis can be computed, we need a generative model. Bayesian approach needs moreover prior information. Nevertheless, even if choosing the ratio $\frac{p_0}{p_1}$ rather than α^* seems more satisfying, we believe that priors remain spoilt by arbitrariness.

What is called in Desolneux *et al.* works *a contrario* model corresponds with this terminology to the modelling of the likelihood of the counter-hypothesis. Precisely speaking in our situation, a curve $\mathcal{C}$ being

given, our hypothesis is $\mathcal{H}_0$ "is an observation consistent with our model for distances to $\mathcal{C}$?" (independence of features). We compute the probability that an observed curve $\mathcal{C}'$ *does not fit* this model, *i.e.* $\mathcal{L}(d(\mathcal{C}', \mathcal{C})|\mathcal{H}_1)$ (where $d(\mathcal{C}', \mathcal{C})$ stands for the list of distances between features). This likelihood distribution is built up thanks to the *background model*, which is trained on a database. The proposed decision rule is the following: if $\mathcal{L}(d(\mathcal{C}', \mathcal{C})|\mathcal{H}_1) < \varepsilon/N$ (where $N = \#\Omega$ is the number of observations), then we reject the $\mathcal{H}_1$ hypothesis and we classify $\mathcal{C}'$ in Ω_0 (that is to say we decide that $\mathcal{C}'$ matches $\mathcal{C}$).

The two key points of our method are:

- We prove that the probability of false alarm α is then bounded by ε/N, what gives an accurate meaning to the only parameter of this test. Since the background model allows $P(\mathcal{C}'|\mathcal{H}_1)$ to reach very low numerical values (typically circa 10^{-6}), being very strict with ε is possible. This point is an improvement with regards to classical statistical tests for which α is set to 1% or 5%.

- We do not control explicitly the probability of non-detection α_2. Nevertheless, experimental evidences show that α_2 is quite low. Moreover, empirical misses come mostly from the fact that one of the shapes has not been correctly extracted. This is another problem. Of course, we observe a trade-off between α and α_2: if ε increases, so do both α and α_2.

The strength of the method we proposed is that we only need a background model (that consequently makes detections relative to the searched curve and to the context); there is no need for the likelihood model of the hypothesis. Moreover, we do not need any prior information. We believe that the proposed method, which detects structures plunged in a background (provided that enough observations are available to estimate this background) is quite general and is certainly the only possible decision rule in most realistic situations, where likelihood model would mean generative model, and where no clear prior information is available.

We could find it quite paradoxical to model the background model instead of the variations of the sought shape. Nevertheless, most authors model the shapes because they actually deal with *object recognition*, and not with *shape detection* in a general manner. Object recognition theory assumes that objects are well extracted from images (*e.g.* contours from well contrasted tools): the sought shape is here a variation around a small number of pre-defined models (which allows to build likelihood-functions), even priors can be defined, and since the models are different enough, the likelihoods are well separated, what makes the decision easier. As far as we are concerned, the point of interest is to decide to which shape in a database does match a sought shape. We expect nothing particular about the sought shapes. We therefore have no other choice but to compare the shape of interest to the only knowledge of the world we get: the background model. The success of our model reposes on the fact that most human-made shapes present structures that make them really special with regards to background shapes generated 'just by chance', and can be finally easily discriminated.

8 Perspectives and conclusion

The aim of this article is to propose a method to compute the Number of False Alarms of a match between some pieces of level lines, up to a similitude invariance. Computing this quantity is useful because it leads to an acceptance / rejection threshold for shape matching. The proposed decision rule is to keep in consideration the matches with a NFA lower than 10^{-1}. Such a match should not appear in a white noise image, and should therefore be considered as highly relevant.

Of course, dealing only with pieces of level lines is not enough to decide whether an object is present, or not, in a given image. Nevertheless, object edges coincide well with pieces of level lines. Future research will thus combine the matches, by taking account of their spatial coherence. Since groups of matches should have a still lower NFA than a single match, the acceptation / rejection threshold will be all the stronger, as noticed on preliminary results.

Acknowledgements: This work was supported by the Office of Naval Research under grant N00014-97-1-0839, by the Centre National d'Études Spatiales, and by the Réseau National de Recherche en Télécommunications (projet ISII).

References

[1] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):502–507, 2003.

[2] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing: Multiscale analysis and P.D.E. *Archive for Rational Mechanics and Analysis*, 16(9):200–257, 1993.

[3] N. Arnaud, F. Cavalier, M. Davier, and P. Hello. Detection of gravitational wave bursts by interferometric detectors. *Physical review D*, 59(8):082002–1 – 082002–9, 1999.

[4] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[5] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using the Expectation-Maximization algorithm and its application to content-based image retrieval. In *Proceedings of the International Conference on Computer Vision*, pages 675–682, Mumbai, India, 1998.

[6] F. Cao. Contrast invariant detection of good continuations, corners and terminators. Technical report, Irisa, 2003.

[7] F. Cao, P. Musé, and F. Sur. Extracting meaningful curves from images. Submitted to Journal of Mathematical Imaging and Vision, 2004.

[8] P.B. Chapple, D.C. Bertilone, R.S. Caprari, and G.N. Newsam. Stochastic model-based processing for detection of small targets in non-gaussian natural imagery. *IEEE Transactions on Image Processing*, 10(4):554–564, 2001.

[9] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

[10] A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.

[11] P.A. Devijver and J. Kittler. *Pattern recognition - A statistical approach.* Prentice Hall, 1982.

[12] S.A. Dudani, K.J. Breeding, and R.B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, 26(1):39–46, 1977.

[13] U. Grenander. *General pattern recognition.* Oxford Science Publications, 1993.

[14] W.E.L. Grimson and D.P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1201–1213, 1991.

[15] E.T. Jaynes. *Probability theory - the logic of science.* Cambridge University Press, 2003.

[16] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 335–344, Ann Arbor, Michigan, U.S.A., 1988.

[17] Y. Lamdan and H.J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 238–249, Tampa, Florida, USA, 1988.

[18] J.-L. Lisani. *Shape Based Automatic Images Comparison.* Thèse de doctorat, Université Paris 9 Dauphine, France, 2001.

[19] J.-L. Lisani, L. Moisan, J.-M. Morel, and P. Monasse. On the theory of planar shape. *SIAM Multiscale Modeling and Simulation*, 1(1):1–24, 2003.

[20] L. Moisan. Affine plane curve evolution: A fully consistent scheme. *IEEE Transactions on Image Processing*, 7(3):411–420, 1998.

[21] P. Monasse. *Représentation morphologique d'images numériques et application au recalage.* Thèse de doctorat, Université Paris 9 Dauphine, France, 2000.

[22] P. Monasse and F. Guichard. Fast computation of a contrast invariant image representation. *IEEE Transactions on Image Processing*, 9(5):860–872, 2000.

[23] P. Musé, F. Sur, F. Cao, and Y. Gousseau. Unsupervised thresholds for shape matching. In *Proceedings of the IEEE International Conference on Image Processing*, Barcelona, Spain, 2003.

[24] P. Musé, F. Sur, and J.-M. Morel. Sur les seuils de reconnaissance des formes. *Traitement du Signal*, 20(3):279–294, 2003.

[25] C. Olson and D.P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(12):103–113, 1997.

[26] B.L. Raktoe and J.J. Hubert. *Basic Applied Statistics*. Marcel Dekker Inc., 1979.

[27] C.A. Rothwell. *Object Recognition Through Invariant Indexing*. Oxford Science Publications, 1995.

[28] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11(1):25–44, 1993.

[29] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[30] S.D. Silvey. *Statistical Inference*. Chapman and Hall, 1975.

[31] C.G. Small. *The Statistical Theory of Shapes*. Springer, 1996.

[32] R. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. In M.S. Lew, editor, *Principles of Visual Information Retrieval*, volume XIX. Springer, 2001.

[33] G.H. Watson and S.K. Watson. Detection of unusual events in intermittent non-gaussian images using multiresolution background models. *Optical Engineering*, 35(11):3159–3171, 1996.

[34] A. Winter and C. Nastar. Differential feature distribution maps for image segmentation and region queries in image databases. In *CBAIVL Workshop at Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, 1999.

[35] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3):269–281, 1972.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399